

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-134203

(43)Date of publication of application : 21.05.1999

(51)Int.Cl.

G06F 9/46

G06T 1/00

H04N 7/24

(21)Application number : 10-238057

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 25.08.1998

(72)Inventor : IMANISHI HIROSHI
ARAKI TOSHIYUKI

(30)Priority

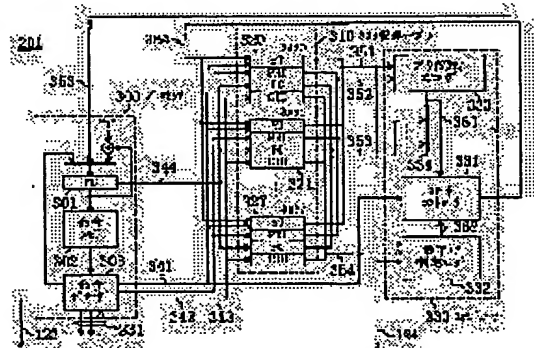
Priority number : 09235625 Priority date : 01.09.1997 Priority country : JP

(54) MICRO CONTROLLER, DATA PROCESSING SYSTEM AND CONTROL METHOD FOR TASK SWITCH

(57)Abstract:

PROBLEM TO BE SOLVED: To realize high speed task switching in a micro controller for controlling plural hardware engines.

SOLUTION: A processor 300, a task management table 310 and a scheduler 330 are incorporated in the micro controller. The processor 300 sequentially executes plural tasks for controlling the allocated hardware engines (cores). The task management table 310 stores task management information containing state information (ST information) showing the respective execution situations of the plural tasks, priority information (PRI information) showing the execution priority of the plural tasks and core ID information (CID information) showing to which cores the plural tasks are allocated. The scheduler 330 causes the processor 300 to switch the task based on task management information when a specified instruction is decoded or the execution of any core terminates.



LEGAL STATUS

[Date of request for examination] 25.08.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3007612

[Date of registration] 26.11.1999

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

THIS PAGE BLANK (USPTO)

THIS PAGE BLANK (USPTO)

THIS PAGE BLANK (USPTO)

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (J P) (12) 特 許 公 報 (B 2)

(11) 特許番号
特許第3007612号
(P3007612)

(45) 発行日 平成12年2月7日(2000.2.7)

(24) 登録日 平成11年11月26日(1999.11.26)

(51) Int. Cl. ⁷	識別記号	P 1
G 0 6 F	9/46	3 4 0 E
G 0 6 T	1/00	3 4 0 B
H 0 4 N	7/24	15/68
		J
		Z

請求項の数18(全 11 項)

(21) 出願番号	特願平10-228057	(73) 特許権者	000005621 松下電器産業株式会社
(22) 出願日	平成10年8月25日(1998.8.25)	(72) 発明者	大坂府門真市大字門真1006番地 今西 博
(56) 公開番号	特開平11-134283		大坂府門真市大字門真1006番地 松下電 器産業株式会社内
(43) 公開日	平成11年5月21日(1999.5.21)	(72) 発明者	荒木 敏之
(31) 優先権主張番号	特願平9-235625		大坂府門真市大字門真1006番地 松下電 器産業株式会社内
(32) 優先日	平成9年9月1日(1997.9.1)	(74) 代理人	100077931 弁理士 前田 弘 (外2名)
(33) 優先権主張国	日本 (J P)	審査官	久保 光宏

最終頁に続く

(54) 【発明の名称】 マイクロコントローラ、データ処理システム及びタスクスイッチの制御方法

(57) 【特許請求の範囲】
1
【請求項1】 複数のハードウェアエンジンと協働して、かつプログラムされた命令に従って、複数のタスクをシーケンシャルに実行するためのプロセッサと、前記複数のタスクの各々の実行状況を表すステータス情報と、前記複数のタスクの各々の実行優先度を表すプライオリティ情報と、前記複数のタスクの各々がいつれのハードウェアエンジンに割り当てられているかを表す割り当て情報とを含むタスク管理情報を記憶するためのタスク管理テーブルと、
前記タスク管理情報に基づいて前記プロセッサにタスクスイッチをさせるためのスケジューラとを備え、
前記複数のハードウェアエンジンの各々は、前記プロセッサによる起動に従ってデータ処理の実行を開始し、かつ前記データ処理の実行が終了すると前記スケジューラに

2

対して実行終了を知らせ、
前記スケジューラは、いずれかのハードウェアエンジンの実行終了を検出した場合には前記プロセッサにタスクスイッチをさせ、
前記複数のタスクの処理を、その内容に応じて、前記プロセッサによりシーケンシャルに実行される部分と、前記複数のハードウェアエンジンにより実行される部分とに分割する構成としたことを特徴とするマイクロコントローラ。

10

【請求項2】 請求項1記載のマイクロコントローラにおいて、
前記複数のタスクの各々は、実行待ちを表す第1のステータスと、実行中を表す第2のステータスと、割り当てられたハードウェアエンジンの実行終了待ちを表す第3のステータスとを有することを特徴とするマイクロコントローラ。

3

【請求項3】 請求項2記載のマイクロコントローラにおいて、
前記プロセッサは、実行中のタスクにおいて該タスクに割り当てられたハードウェアエンジンを起動した後に特定の命令をデコードしたとき、該タスクのステータスを前記第2のステータスから前記第3のステータスへ変更するように前記タスクの情報更新する機能を有することを特徴とするマイクロコントローラ。

【請求項4】 請求項2記載のマイクロコントローラにおいて、
前記スケジューラは、
いずれかのハードウェアエンジンが実行終了したとき、該ハードウェアエンジンに割り当てられたタスクを前記タスク管理情報に基づいて判定するための判定ユニットと、
前記判定ユニットにより起動されたとき、前記判定されたタスクのステータスを前記第3のステータスから前記第1のステータスへ変更するように前記ステータス情報を更新する機能を有するマイクロコントローラとを備えたことを特徴とするマイクロコントローラ。

【請求項5】 請求項4記載のマイクロコントローラにおいて、
前記ステータスコントローラは、前記判定ユニットにより起動されたとき、実行中のタスクのステータスを前記第2のステータスから前記第1のステータスへ変更するように前記ステータス情報を更新する機能を有するマイクロコントローラとを備えたことを特徴とするマイクロコントローラ。

【請求項6】 請求項2記載のマイクロコントローラにおいて、
前記スケジューラは、前記タスク管理情報に基づき、前記第1のステータスにあるタスクのうち最も高い実行優先度を有するタスクを次に実行すべきタスクとして選択するためのプライオリティエンコーダを更に備えたことを特徴とするマイクロコントローラ。

【請求項7】 請求項6記載のマイクロコントローラにおいて、
前記ステータスコントローラは、前記プライオリティエンコーダにより選択されたタスクのステータスを前記第1のステータスから前記第2のステータスへ変更するように前記ステータス情報を更新する機能を更に有することを特徴とするマイクロコントローラ。

【請求項8】 請求項1記載のマイクロコントローラにおいて、
前記タスク管理テーブルは、前記タスクスイッチの時点より前に実行されていたタスクに係る前記プロセッサの資源を迅速する領域を有することを特徴とするマイクロコントローラ。

【請求項9】 請求項1記載のマイクロコントローラにおいて、

(2)

前記複数のハードウェアエンジンの各々が互いに独立した作業領域として使用する複数のレジスタファイルを更新したことを特徴とするマイクロコントローラ。
【請求項10】 請求項1記載のマイクロコントローラにおいて、
前記複数のハードウェアエンジンのうちの少なくとも2個のハードウェアエンジンに共通の既定バリエータを記憶するためのレジスタファイルを更に備えたことを特徴とするマイクロコントローラ。

【請求項11】 各々データ処理を実行するための複数のハードウェアエンジンと、
前記複数のハードウェアエンジンを制御するためのマイクロコントローラとを備えたデータ処理システムであって、
前記マイクロコントローラは、
前記複数のハードウェアエンジンと協働して、かつプログラムされた命令に従って、複数のタスクをシーケンシャルに実行するためのプロセッサと、
前記複数のタスクの各々の実行状況を表すステータス情報と、前記複数のタスクの各々の実行優先度を表すプライオリティ情報と、前記複数のタスクの各々がいつれのハードウェアエンジンに割り当てられているかを表す割り当て情報とを含むタスク管理情報を記憶するためのタスク管理テーブルと、
前記タスク管理情報に基づいて前記プロセッサにタスクスイッチをさせるためのスケジューラとを備え、
前記複数のハードウェアエンジンの各々は、前記プロセッサによる起動に従ってデータ処理の実行を開始し、かつ前記データ処理の実行が終了すると前記スケジューラに

対して実行終了を知らせ、
前記スケジューラは、いずれかのハードウェアエンジンの実行終了を検出した場合には前記プロセッサにタスクスイッチをさせ、
前記複数のタスクの処理を、その内容に応じて、前記プロセッサによりシーケンシャルに実行される部分と、前記複数のハードウェアエンジンにより実行される部分とに分割する構成としたことを特徴とするデータ処理システム。

【請求項12】 請求項11記載のデータ処理システムにおいて、
前記複数のタスクの各々は、実行待ちを表す第1のステータスと、実行中を表す第2のステータスと、割り当てられたハードウェアエンジンの実行終了待ちを表す第3のステータスとを有することを特徴とするデータ処理システム。

【請求項13】 請求項11記載のデータ処理システムにおいて、
前記複数のハードウェアエンジンは、各々MPEG画像データのエンコードのための部分処理コアであることを特徴とするデータ処理システム。

4

前記複数のハードウェアエンジンの各々が互いに独立した作業領域として使用する複数のレジスタファイルを更新したことを特徴とするマイクロコントローラ。

【請求項10】 請求項1記載のマイクロコントローラにおいて、
前記複数のハードウェアエンジンのうちの少なくとも2個のハードウェアエンジンに共通の既定バリエータを記憶するためのレジスタファイルを更に備えたことを特徴とするマイクロコントローラ。

【請求項11】 各々データ処理を実行するための複数のハードウェアエンジンと、
前記複数のハードウェアエンジンを制御するためのマイクロコントローラとを備えたデータ処理システムであって、
前記マイクロコントローラは、
前記複数のハードウェアエンジンと協働して、かつプログラムされた命令に従って、複数のタスクをシーケンシャルに実行するためのプロセッサと、
前記複数のタスクの各々の実行状況を表すステータス情報と、前記複数のタスクの各々の実行優先度を表すプライオリティ情報と、前記複数のタスクの各々がいつれのハードウェアエンジンに割り当てられているかを表す割り当て情報とを含むタスク管理情報を記憶するためのタスク管理テーブルと、
前記タスク管理情報に基づいて前記プロセッサにタスクスイッチをさせるためのスケジューラとを備え、
前記複数のハードウェアエンジンの各々は、前記プロセッサによる起動に従ってデータ処理の実行を開始し、かつ前記データ処理の実行が終了すると前記スケジューラに

対して実行終了を知らせ、
前記スケジューラは、いずれかのハードウェアエンジンの実行終了を検出した場合には前記プロセッサにタスクスイッチをさせ、
前記複数のタスクの処理を、その内容に応じて、前記プロセッサによりシーケンシャルに実行される部分と、前記複数のハードウェアエンジンにより実行される部分とに分割する構成としたことを特徴とするデータ処理システム。

【請求項12】 請求項11記載のデータ処理システムにおいて、
前記複数のタスクの各々は、実行待ちを表す第1のステータスと、実行中を表す第2のステータスと、割り当てられたハードウェアエンジンの実行終了待ちを表す第3のステータスとを有することを特徴とするデータ処理システム。

【請求項13】 請求項11記載のデータ処理システムにおいて、
前記複数のハードウェアエンジンは、各々MPEG画像データのエンコードのための部分処理コアであることを特徴とするデータ処理システム。

5

【請求項14】 少なくとも1つのタスクがそれぞれ対応するハードウェアエンジンに割り当てられ、その割り当てを表す情報に基づいてタスクジェネラータでタスクスイッチを制御するタスクスイッチの制御方法であって、タスクスイッチの処理その内容に応じて、プロセスによりタスクジェネラルに表される部分と、複数のハードウェアエンジンにより実行される部分とに分割し、各ハードウェアエンジンは、前記プロセスによる分割に従ってデータ処理の実行を開始し、かつ数データ処理の実行が終了すると前記タスクジェネラルに対して実行終了を知らせる。

各タスクは、実行待ちを表す第1のステートと、実行中を表す第2のステートと、割り当てられたハードウェアエンジンの実行終了待ちを表す第3のステートとを有し、前記タスクジェネラルは、いずれかのハードウェアエンジンが実行終了したとき、前記プロセスにタスクスイッチをさせるように、実行終了したハードウェアエンジンに割り当てられたタスクのステートを前記第3のステートから前記第1のステートへ変更することを特徴とするタスクスイッチの制御方法。

【請求項15】 請求項14記載のタスクスイッチの制御方法において、ハードウェアエンジンが実行終了したとき、実行中のタスクのステートを前記第2のステートから前記第1のステートへ変更することを特徴とするタスクスイッチの制御方法。

【請求項16】 複数のハードウェアエンジンと接続して、かつプログラムされた命令に従って、複数のタスクをタスクジェネラルに実行するためのプロセスと、前記複数のタスクの各々の実行優先度を表すプライオリティ情報と、前記複数のタスクの各々がいずれのハードウェアエンジンに割り当てられているかを表す割り当て情報とを含むタスク管理情報を記憶するためのタスク管理テーブルと、前記タスク管理情報に基づいて前記プロセスにタスクスイッチをさせるためのタスクジェネラルと、前記複数のハードウェアエンジンに共通の数値パラメータを記憶するためのレジスタファイルとを備えたことを特徴とするマイクロコントローラ。

【発明の詳細な説明】
【0001】
【発明の属する技術分野】 本発明は、マルチタスク環境を備えたマイクロコントローラと、該マイクロコントローラが複数のハードウェアエンジンとを制御するように構成されたデータ処理システムとに関するものである。また、本発明はタスクスイッチの制御方法に関するものである。

6

【0002】
【従来の技術】 マルチタスク環境を備えたマイクロコントローラが知られている。この種のマイクロコントローラに内蔵された単一のプロセッサは、複数のタスクをタスクジェネラルに実行する。そのため、タスクスイッチを要求するタイマ割り込みをタスクスイッチが定期的に発行する。このタイマ割り込みをプロセスが受理するたびに、オペレーティングシステム (OS) 内の割り込み処理ルーチンが起動され、割り込み処理ルーチンがタスクスイッチジェネラルと資源の遅延及び復旧とを行う。

【0003】
【発明が解決しようとする課題】 上記従来のマイクロコントローラは、割り込み処理ルーチンでタスクスイッチジェネラルを行っていたことから、タスクスイッチにおけるオーバーヘッドが大きくなり、マイクロコントローラの実質的な稼働率が低下するという問題があった。これは、特に画像データのエンコードのようなリアルタイム性を重視するアプリケーションにおいて深刻な問題である。

【0004】 本発明の目的は、マイクロコントローラにおける高速のタスクスイッチを実現することにある。

【0005】 本発明他の目的は、マイクロコントローラが複数のハードウェアエンジンを制御するように構成されたデータ処理システムにおいて、該マイクロコントローラにおける高速のタスクスイッチを実現することにある。

【0006】 本発明更に他の目的は、高速のタスクスイッチを実現するためのタスクスイッチの制御方法を提供することにある。

【0007】
【課題を解決するための手段】 上記目的を達成するため、本発明のマイクロコントローラは、割り込み処理ルーチンでタスクスイッチを制御するのではなく、当該マイクロコントローラ中のプロセスが複数のハードウェアエンジンと協調してかつプログラムされた命令に従って複数のタスクをタスクジェネラルに実行し、かつ複数のタスクがそれぞれ対応するハードウェアエンジンに割り当てられた状態で、その割り当てを表す情報に基づいてハードウェアタスクジェネラルでタスクスイッチを制御することとしたものである。複数のハードウェアエンジンの各々は、プロセスにより起動に従ってデータ処理の実行を開始し、かつ数データ処理の実行が終了するとタスクジェネラルに対して実行終了を知らせる。タスクジェネラルは、いずれかのハードウェアエンジンにタスクスイッチを渡した場合には、プロセスにタスクスイッチをさせる。しかるに、複数のタスクの処理を、その内容に応じて、プロセスによりタスクジェネラルにより実行される部分と、複数のハードウェアエンジンにより実行される部分とに分割する構成とした。複数のハードウェアエンジンの中には時間的にクリティカルな処理を実行するもの

7

と、そうでないものがある。本発明によれば、このような複数のハードウェアエンジン間の関係が複数のタスクの各々の実行優先度に反映される結果、いずれのハードウェアエンジンが時間的にクリティカルな処理を実行するかをタスクスイッチ時にあらかじめ判断することなく、次に実行すべきタスクを短時間のうちに選択できる。つまり、タスクスイッチにおけるオーバーヘッドが小さくなり、高速のタスクスイッチが実現される。

【0008】 また、本発明のマイクロコントローラでは、タスクスイッチから定期的に発行される割り込みに応じてタスクスイッチを行う時分方式では無待時間が生じることになり、各ハードウェアエンジンの実行終了というイベントの発生に直ちにตอบสนองしてタスクスイッチを行うイベントドリブン方式を採用した。複数のタスクの各々は、実行待ちを表す第1のステート (READYステート) と、実行中を表す第2のステート (ACTIVEステート) と、割り当てられたハードウェアエンジンに割り当てられたタスクのステート (SLEEPステート) とを有する。ACTIVEステートは、現在、タスクがマイクロコントローラを使用している状態であり、そのタスクに割り当てられたハードウェアエンジンの制御を行う。READYステートは、タスクがマイクロコントローラを使用可能な状態ではあるが、そのタスクは選ばれておらず、選ばれたのを待っている状態である。SLEEPステートは、割り当てられたハードウェアエンジンに割り当てられたタスクは、タスクがマイクロコントローラを使用可能な状態ではあるが、そのタスクは選ばれておらず、選ばれたのを待っている状態である。

特定の命令 (task sleep 命令) に応じて、ACTIVEステートからSLEEPステートへ遷移する。あるハードウェアエンジンが実行終了したとき、該ハードウェアエンジンに割り当てられたタスクはSLEEPステートからREADYステートへ遷移し、実行中のタスクのうちで最も高い実行優先度を有するタスクが次に実行すべきタスクとして選択され、該選択されたタスクがREADYステートからACTIVEステートへ遷移する。

【0009】 複数のハードウェアエンジンの各々が互いに独立した作業領域として使用する複数のレジスタファイルとマイクロコントローラの中に用意しておけば、タスクスイッチ時にはプログラムカウンタなどのプロセス資源のみを遷移すればよく、タスクスイッチにおけるオーバーヘッドが更に小さくなる。複数のハードウェアエンジンに共通の数値パラメータを記憶するためのレジスタファイルはマイクロコントローラの中に用意しておいてもよい。

【0010】
【発明の実施の形態】 図1は、本発明に係るデータ処理

8

システムの1つであるMPPEG (Moving Picture Experts Group) 画像エンコーダの構成例を示している。図1のエンコーダは、単一のマイクロコントローラ101と、マイクロコントローラ101を構成する5個のハードウェアエンジン (以下、コアという) 111~115と、3個のバッファメモリ116~118とで構成されている。5個のコアは、動き検出器 (Motion Detector: MD) 111、動き補償器 (Motion Compensator: MC) 112、離散コサイン変換器 (Discrete Cosine Transformer: DCT) 113、量子化器 (Quantizer: Q) 114及び可変長符号化器 (Variable Length Code: VLC) 115であって、マルチタスク環境を備えたマイクロコントローラ101によりそれぞれ制御される。121はエンコーダ101にそれぞれ割り当てられたエンコード結果を表すデータである。マイクロコントローラ101は、5個のコア111~115の各々へ起動信号123を供給し、かつ5個のコア111~115の各々から終了信号124を受け取る。また、マイクロコントローラ101は、信号線131~135を介して5個のコア111~115と個別に通信したり、信号線136を介して5個のコア111~115に共通のパラメータを与えたりすることができるように構成されている。

【0011】 図2は、マイクロコントローラ101の詳細構成を示している。マイクロコントローラ101は、マルチタスク環境を実現するためのタスクコントローラ201と、上記5個のコア111~115の各々が互いに独立した作業領域として使用する5個のコアレジスタファイル211~215と、上記5個のコア111~115の各々の少なくとも2個のコアに共通の数値パラメータを記憶するための1個の共通レジスタファイル216と、タスクコントローラ201が作業領域として使用する1個の汎用レジスタファイル217と、乗算器221と、シフト222と、算術論理演算ユニット (Arithmetic and Logic Unit: ALU) 223と、データメモリ224とを備えている。241はAバス、242はBバス、243はCバス、231はこれらのバスとタスクコントローラ201とを接続するための信号線である。タスクコントローラ201は、上記起動信号123を供給し、かつ上記終了信号124を受け取る。レジスタファイル211~216の各々は、Cバス243と上記信号線131~136のうちの対応する信号線との間に存在し、かつその各々の出力がAバス241及びBバス242にそれぞれ接続されている。汎用レジスタファイル217及びデータメモリ224の各々の出力はCバス243に、その各々の2出力はAバス241及びBバス242にそれぞれ接続されている。乗算器221、シフト222及びALU223の各々の出力はAバス241及びBバス242に、その各々の出力はCバス243にそれぞれ接続されている。なお、5個のコアレジ

9

スタタアル211～215及び共通レジスタアル216の配線を省略して、上記信号線131～136をCバス243から直接に引き出すようにしてもよい。

[0012] 図1のMP E G画像エンコーダによれば、16×16画素からなるブロックを単位として画像データ処理が進められる。まず、入力された画像データ121に対してMDコプ111で動きベクトルの検出が求められる。これらの動きベクトルを用いてMCコプ112で画像の差分データが求められ、最適な動きベクトルが選択される。選択された動きベクトルに対する差分データは、DCTコプ113で離散コサイン変換され、Qコプ114で量子化され、求めた動きベクトルなどのサイド情報とともにVLCコプ115で可変長符号化された後、符号化データ122として出力される。

[0013] 図2を参照して詳細に説明すると、タスクコントローラ201は、まず信号線231、ALU223及びCバス243を介してMDコプレジスタアル211に動作パラメータを設定し、起動信号123によりMDコプ111を起動する。MDコプ111は、MDコプレジスタアル211から信号線131を介して動作パラメータを読み込むとともに、画像データ121を入力する。MDコプ111の実行が終了すると、求められた動きベクトルの検出が信号線131を介してMDコプレジスタアル211に書き込まれ、MDコプ111から終了信号124が出力される。タスクコントローラ201は、この終了信号124を受け取り、MDコプレジスタアル211から動きベクトルの検出を読み出し、これに基づき乗算器221、シフト222、ALU223及び汎用レジスタアル217を用いてMCコプ112のための動作パラメータを計算する。この動作パラメータはMCコプレジスタアル212に設定され、起動信号123によりMCコプ112が起動される。MCコプ112は、MCコプレジスタアル212から信号線132を介して動作パラメータを読み込み、その後、画像の差分データを求める。MCコプ112の実行が終了すると、差分データの検出が信号線132を介してMCコプレジスタアル212に、画像の差分データがバスワメモリ116にそれぞれ書き込まれ、MCコプ112から終了信号124が出力される。タスクコントローラ201は、この終了信号124を受け取り、MCコプレジスタアル212から差分データの検出を読み出し、これに基づき乗算器221、シフト222、ALU223及び汎用レジスタアル217を用いて上記乗算の中から最適な動きベクトルを選択する。求められた動きベクトルに対する差分データの検出がDCTコプレジスタアル213に設定される。DCTコプ113は、DCTコプレジスタアル213に設定されたアルゴリズムに基づいてバスワメモリ116から差分データを読み出し、これを離散コサイン

10

変換する。DCTコプ113の実行が終了すると、離散コサイン変換の結果がバスワメモリ117に書き込まれ、DCTコプ113から終了信号124が出力される。以下、Qコプ114で量子化処理がなされ、その結果がバスワメモリ118に書き込まれ、VLCコプ115で可変長符号化処理がなされ、この結果が符号化データ122として出力される。なお、上記5個のコプ111～115のうちのいくつかのコプは、1マクロブロックを処理する間に、マイクロコントローラ101と起動信号123及び終了信号124のやりとりを複数回行う。共通レジスタアル216は、MPEG1とMPEG2との切り替えのための共通パラメータを5個のコプ111～115へ予め供給したり、動き評価モードを指定するための共通パラメータをMDコプ111及びMCコプ112へ予め供給したりする際に用いられる。

[0014] 図3は、タスクコントローラ201の詳細構成を示している。タスクコントローラ201は、プロセッサ300と、タスク管理テーブル310と、スタジアム300とで構成されている。プロセッサ300は、最大8個のタスクをスケジューラに実行することができるRISC (Reduced Instruction Set Computer) 型のプロセッサであって、命令アドレスを生成するためのプログラムカウンタ (PC) 301と、一連の命令からなるプログラムを格納するための命令メモリ302と、命令をフェッチするための命令デコーダ303とを備えている。各コプへの起動信号123は、命令デコーダ303から供給される。また、命令デコーダ303は、命令実行のための資源である上記乗算器221、シフト222、ALU223などに信号線231を介して接続されている。タスク管理テーブル310は、タスク管理情報を記憶するための回路ブロックであって、タスク40からタスク7までの8個のタスクのそれぞれに対応した8個のタスク管理レジスタアル320～327を備えている。ここに、タスク管理情報は、複数のタスクの各々の実行状況を表すステータス情報 (ST情報) と、該複数のタスクの各々の実行優先度を表すプライオリティ情報 (PRI情報) と、該複数のタスクの各々が5個のコプ111～115のうちのいずれのコプに割り当てられているかを表すコプID情報 (CID情報) とを含む。更に、タスク管理テーブル310は、プロセッサ300の資源の有無を示すPFC301の内容を記憶するためのタスク毎の領域を有する。この記憶領域は、ALU223 (図2参照) の演算結果に係るフラグなども含まれる。スケジューラ330は、タスク管理テーブル310に記憶されたタスク管理情報に基づいてプロセッサ300にタスクを実行させるための回路ブロックであって、ステータスコントローラ331と、終了コプ判定ユニット332と、プライオリティエンコーダ333と、セレクトラ334とを備えている。終了コプ判定ユニット332は、5個のコプ111～115のうちのい

11

れのコプから終了信号124を受け取ったとき、実行終了したコプに割り当てられたタスクを判定するためのユニットである。この判定はタスク管理テーブル310を参照して行われ、判定結果を表すタスク番号362がステータスコントローラ331に知られる。プライオリティエンコーダ333は、次に実行すべきタスクを選択するための回路ブロックである。この選択はタスク管理テーブル310を参照して行われ、選択結果を表すタスク番号361がステータスコントローラ331及びセレクトラ334に知られる。ステータスコントローラ331は、タスク管理テーブル310の中のST情報を更新するための回路ブロックである。セレクトラ334は、プロセッサ300への資源確保を行う。

[0015] 図4は、図1のMP E G画像エンコーダにおけるコプとタスクとの対応関係を示している。マイクロコントローラ101は、ここでは6個のタスク400～405を実行する。タスク400は、下位階層の5個のタスク401～405を制御し、かつエンコード処理の全体を管理するためのメインタスク (タスク40) である。このメインタスク400が割り当てられるべきコプ111を制御するための動き補償タスク (タスク112) を制御する。タスク402は、割り当てられたMCコプ112を制御するための動き補償タスク (タスク2) である。タスク403は、割り当てられたDCTコプ113を制御するための離散コサイン変換タスク (タスク3) である。タスク404は、割り当てられたQコプ114を制御するための量子化タスク (タスク4) である。タスク405は、割り当てられたVLCコプ115を制御するための可変長符号化タスク (タスク5) である。

[0016] ここで、図4に示す少なくとも6個のタスク400～405に係るタスク管理情報が図3中のタスク管理テーブル310に記憶されているものとする。図3によれば、PRI情報はプライオリティ設定信号342に応じて、CID情報はコプ設定信号343に応じてそれぞれ設定される。プライオリティ設定信号342は命令デコーダ303がプライオリティ設定命令をデコードした際に、コプ設定信号343は命令デコーダ303がコプ設定命令をデコードした際にそれぞれ命令デコーダ303からタスク管理テーブル310へ供給される。

[0017] 図5は、各タスクのステータスへ遷移を示す概略図である。タスクは、停止を表すSTOPステータスと、実行待ちを表すREADYステータスと、実行中を表すACTIVEステータスと、割り当てられたコプの実行終了待ちを表すSLEEPステータスを有する。ただし、タスク40にはSLEEPステータスがない。リセット直後のタスクはSTOPステータスにある。STOPステータスにあるタスクは、task_ready命令によりREADYステータスへ遷移せられる (遷移501)。

12

要求するイベントの発生時にタスク330により選択されると、ACTIVEステータスへ遷移せられる (遷移511)。この際、その時点でACTIVEステータスにあったタスクは、スケジューラ330によりREADYステータスへ遷移せられる (遷移522)。ACTIVEステータスにあるタスクは、プロセッサ300により実行され、task_sleep命令によりSLEEPステータスへ遷移せられる (遷移521)。task_sleep命令によりSTOPステータスへ遷移せられる (遷移523) することができる。SLEEPステータスにあるタスクは、割り当てられたコプの実行終了によりREADYステータスへ遷移せられる (遷移531)。

[0018] ここで、図3のタスクコントローラ201の詳細動作を説明する。命令デコーダ303がtask_ready命令、task_sleep命令又はtask_sleep命令をフェッチすると、タスクスイッチが発生する。例えば、実行中のタスクにおいて該タスクに割り当てられたコプのための動作パラメータの設定と該コプの起動とを終えたタスクは、task_sleep命令によりACTIVEステータスからSLEEPステータスへ遷移せられる。また、5個のコプ111～115のうちのいずれかのコプが実行終了したときにもタスクスイッチが発生する。タスクスイッチ時のタスクコントローラ201の動作シーケンスは、(1) スケジューラの起動、(2) 実行中のタスクの資源の返還、(3) 次に実行するタスクの選択、(4) 返還されていた資源の復帰である。

[0019] まず、命令に基づきタスクスイッチのシーケンスを実行する。

[0020] (A-1) スケジューラの起動

task_ready命令、task_sleep命令又はtask_sleep命令がフェッチされると、命令デコーダ303からステータス変更信号341が出力される。ステータス変更信号341はステータスコントローラ311に入力される。その結果、スケジューラ330が起動される。

[0021] (A-2) 実行中のタスクの資源の返還

ステータス変更信号341はタスク管理テーブル310にも入力され、ST情報が更新される。同時にその時点まで実行されていたタスクのDCT301の値が、信号線344を介してレジスタ管理テーブル310の中に返還される。

[0022] (A-3) 次に実行するタスクの選択

プライオリティエンコーダ333は、タスク管理テーブル310から信号線351を介してST情報を、信号線352を介してPRI情報をそれぞれ受け取り、READYステータスにあるタスクのうち最も高い実行優先度を有するタスクを次に実行すべきタスクとして選択する。この選択の結果を表すタスク番号361は、ステータス

13

トコンローラ331及びセレクタ334に知らされる。

【0023】(A-4) 遅延されていた資源の復帰
ステートコンローラ331は、タスク番号361に応じたステート変更信号364をタスク管理テーブル310へ供給する。その結果、ブライオリチエシコング33により選択されたタスクのST情報、READYステートからACTIVEステートに更新される。セクタ334は、タスク番号361で指定されたタスクのPCをタスク管理テーブル310から信号線353を介して読み出し、該PCを信号線363へ供給する。その結果、次に実行すべきタスクのPCの値がプロセス300に設定され、該タスクの実行が開始する。

【0024】次に、コアの実行終了に基づくタスクインテグレーションを説明する。

【0025】(B-1) スケジューラの起動
いずれのコアが実行終了すると、終了コア判定ユニット332は、いずれのコアが実行終了したかを終了信号124に基づいて判定する。更に、終了コア判定ユニット332は、いずれのコアが実行終了したかを終了信号124に基づいて判定する。更に、終了コア判定ユニット332は、信号線354を介してタスク管理テーブル310の中のCID情報を読み出し、実行終了したコアにいずれのタスクが割り当てられているかを判定する。この判定の結果をタスク番号362は、ST情報から当該タスクがSLEEPステートにあることが確認された場合に、ステートコンローラ331に知らされる。その結果、スケジューラ330が起動される。ステートコンローラ331は、タスク番号362に応じたステート変更信号364をタスク管理テーブル310へ供給する。その結果、実行終了したタスクのST情報、SLEEPステートからREADYステートに更新される。なお、実行終了したコアに割り当てられたタスクが存在しない場合には、スケジューラ330は起動されない。

【0026】(B-2) 実行中のタスクの資源の遅延
更に、ステートコンローラ331は、その時点まで実行されていたタスクのST情報がACTIVEステートからREADYステートに更新されるように、ステート変更信号364をタスク管理テーブル310へ供給する。同時にその時点まで実行されていたタスクのPC301の値がタスク管理テーブル310の中に遅延される。

【0027】(B-3) 次に実行するタスクの選択
ブライオリチエシコング333は、タスク管理テーブル310からST情報及びPRI情報を取り、READYステートにあるタスクのうちで最も高い実行優先度を有するタスクを次に実行すべきタスクとして選択する。この選択の結果をタスク番号361は、ステートコンローラ331及びセレクタ334に知らされる。

14

【0028】(B-4) 遅延されていた資源の復帰
ステートコンローラ331は、タスク番号361に応じたステート変更信号364をタスク管理テーブル310へ供給する。その結果、ブライオリチエシコング33により選択されたタスクのST情報、READYステートからACTIVEステートに更新される。セクタ334は、タスク番号361で指定されたタスクのPCをタスク管理テーブル310から読み出し、該PCをプロセス300へ供給する。その結果、次に実行すべきタスクのPCの値がプロセス300に設定され、該タスクの実行が開始する。

【0029】図6は、図1中の5個のコア111～115によるマクロブロックパイプライン処理を示している。パイプラインブロックは、各コアにおけるマクロブロックの処理に要する時間の最大値に設定される。したがって、個々のパイプラインブロック間において他のコアより早く実行終了するコアが存在するという特性がある。ここに、アイドル時間が生じる。しかも、アイドル時間の長さが画像データに依存して変化するという特性がある。図1の例では、これらの特性に適合したMPEG画像エンコーダが、イベントドリフ方式のタスクインテグレーションにより実現される。なお、個々のパイプラインブロックの採用により実現される。なお、個々のパイプラインブロック間におけるコアの起動回数は、そのコアで行う処理の内容やデータによって異なる。例えば、DCTコア113は1パイプラインブロック間に1回起動される。MCOコア112は、マクロブロックのデータを輝度成分と色差成分に分け、各成分毎に処理を細分化して実行するため、1パイプラインブロック間にデータに応じて複数回起動される。

【0030】図7は、図6中の縦線で特定された一期間間における3個のタスクの各々のステート遷移の具体例を示している。タスク0はエンコード処理の全体を管理するためのメインタスクであり、タスク1はMDコア111に割り当てられたタスクであり、タスク2はMCOコア112に割り当てられたタスクである(図4参照)。これら3個のタスクのうちタスク1の実行優先度が最も高く、タスク2の実行優先度が2番目に高く、タスク0の実行優先度が最も低いものとする。また、時刻10において、タスク1がACTIVEステートに、タスク0及びタスク2がそれぞれREADYステートにあるものとす。

【0031】図7によれば、時刻11～17の各々においてタスクインテグレーションが発生する。図中のΔは、1回のタスクスイッチにおけるオーバーヘッドを表している。順を追って説明すると、時刻11より前に、タスク1はMDコア111を起動する。そして、時刻11にtask_sleep命令によりタスク1がACTIVEステートからSLEEPステートへ遷移する。この時点ではタスク0及びタスク2がREADYステートにあり、タスク2の実行優先度がタスク0の

15

で、タスク2がREADYステートからACTIVEステートへ遷移する。タスク2はMCOコア112を起動する。そして、時刻12にtask_sleep命令によりタスク2がACTIVEステートからSLEEPステートへ遷移する。この時点ではタスク0のみがREADYステートにあるので、タスク0がREADYステートからACTIVEステートへ遷移する。そして、時刻13にMCOコア112の実行終了によりタスク2がSLEEPステートからREADYステートへ遷移する。また、時刻14にタスク2がREADYステートからACTIVEステートへ遷移する。この時点ではタスク0及びタスク2がREADYステートにあり、タスク2の実行優先度がタスク0の実行優先度より高いので、タスク2がREADYステートからACTIVEステートへ遷移する。タスク2はMCOコア112を再起動する。そして、時刻14にtask_sleep命令によりタスク2がACTIVEステートからSLEEPステートへ遷移する。この時点ではタスク0のみがREADYステートにあるので、タスク0がREADYステートからACTIVEステートへ遷移する。この時点ではタスク0及びタスク1がREADYステートにあり、タスク1の実行優先度がタスク0の実行優先度より高いので、タスク1がREADYステートからACTIVEステートへ遷移する。そして、時刻16にMCOコア112の実行終了によりタスク2がSLEEPステートからREADYステートへ遷移する。これに伴い、その時点までACTIVEステートにあったタスク0がREADYステートへ遷移する。この時点ではタスク0及びタスク1がREADYステートにあり、タスク0の実行優先度がタスク1の実行優先度より高いので、タスク0がREADYステートからACTIVEステートへ遷移する。そして、時刻17にtask_sleep命令によりタスク0がACTIVEステートからSLEEPステートへ遷移する。この時点ではタスク1及びタスク2がREADYステートにあり、タスク1の実行優先度がタスク2の実行優先度より高いので、タスク1がREADYステートからACTIVEステートへ遷移する。タスク1はMDコア111を再起動する。そして、時刻17にtask_sleep命令によりタスク1がACTIVEステートからSLEEPステートへ遷移する。この時点ではタスク0及びタスク2がREADYステートにあり、タスク2の実行優先度がタスク0の実行優先度より高いので、タスク2がREADYステートからACTIVEステートへ遷移する。

【0032】従来の割り込み処理ルーチンを用いた時分方式のタスクスイッチングを採用した場合の1回のタスクスイッチにおけるオーバーヘッドは10数マイクロ秒にもなるが、本発明に係るイベントドリフ方式のタスクスイッチングを採用した場合の図7中のオーバーヘッドは数マイクロ秒にすぎない。図6中の個々のマクロブロック間において最大20回数のタスク

(6)

16

スイッチが発生することを考えると、両方式間のオーバーヘッドの差は更に大きくなる。本発明によるオーバーヘッドの短縮は、パイプラインブロックの短縮をも可能にする。つまり、画像データの高速エンコードが可能になるのである。

【0033】以上のとおり、図1のMPEG画像エンコーダでは高速のタスクスイッチングが実現される。また、時間的にリタイカルな処理を実行するタスクに高い実行優先度を設定しておくことで、正常な画像エンコード処理を確保できる。しかも、いずれのコアが実行終了したとき、該実行終了したコアに割り当てられたタスクのステートと、その時点まで実行中であったタスクのステートをいずれもREADYステートに変更したうえで、READYステートにある全てのタスクのうちで最も高い実行優先度を有するタスクを次に実行すべきタスクとして選択することとしたので、ブライオリチエシコング333の内部構成が簡略化される。更に、各タスク毎にプログラムを独立して記述することができ、で、プログラミンが効率化され、デバッグにも有利である。

【0034】なお、本発明は画像データなどの他のデータ処理システムにも適用可能である。上記の例では全てのマクロエンジン(コア)にタスクを1個ずつ割り当てたが、タスクが割り当てられないコアがあってもよい。複数個のタスクを1個のコアに割り当ててもよい。1個のタスクが複数個のコアに同時に割り当てられることはない。

【0035】
【発明の効果】以上説明してきたとおり、本発明によれば、複数のタスクがそれぞれ対応するハードウェアエンジン(コア)に割り当てられた環境下で、その割り当てを渡す情報に基づいてハードウェアスケジューラでタスクスイッチングを制御することとしたので、高速のタスクスイッチングを実現することができる。

【図面の簡単な説明】

【図1】本発明に係るMPEG画像エンコーダの構成例を示すブロック図である。

【図2】図1中のマクロコンローラの詳細構成を示すブロック図である。

【図3】図2中のタスクコンローラの詳細構成を示すブロック図である。

【図4】図1のエンコーダにおけるコアとタスクとの対応関係を示す概略図である。

【図5】図1中のエンコーダにおけるタスクのステート遷移を示す概略図である。

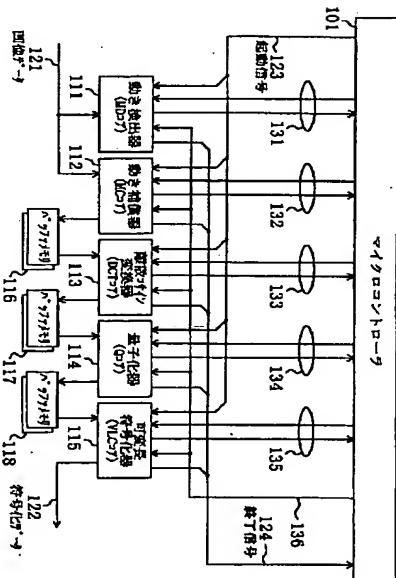
【図6】図1中の5個のコアによるマクロブロックパイプライン処理を示すタイムミング図である。

【図7】図6中の一期間間における3個のタスクの各々のステート遷移の具体例を示すタイムミング図である。

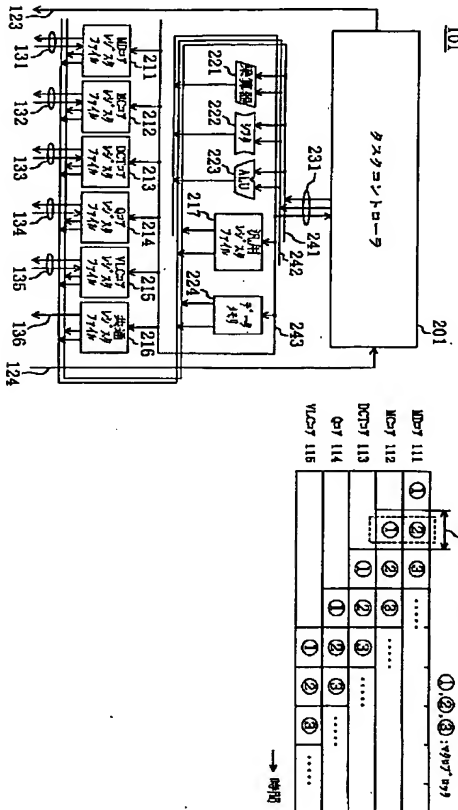
【符号の説明】

- 17
- 101 マイタコンローラ
- 111 動き検出器 (MDコフ)
- 112 動き補償器 (MCコフ)
- 113 離散コサイン変換器 (DCTコフ)
- 114 量子化器 (Qコフ)
- 115 可変長符号化器 (VLCコフ)
- 116~118 パラメータメモリ
- 201 タスクコントローラ
- 18
- * 211~215 コレクションタスク
- 216 共通タスク
- 300 プロセッサ
- 310 タスク管理テーブル
- 330 スケジューラ
- 331 スタータコントローラ
- 332 終了コマンド決定ユニット
- 333 フライオリティエンコーダ

【図1】



【図2】

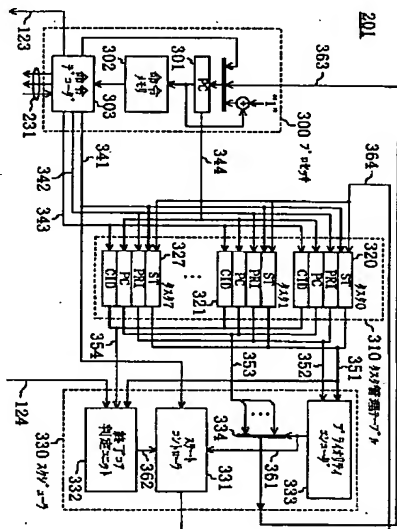


【図6】

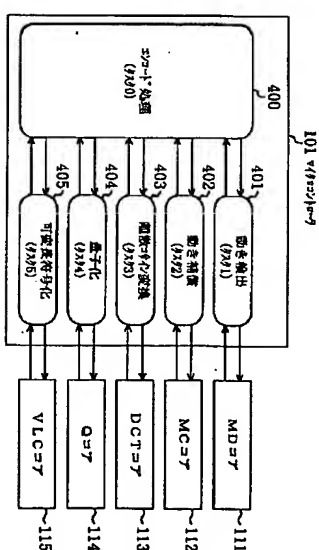
タスク管理テーブル

タスク名	優先度	実行時間	実行回数	実行時刻
タスク1	1	10ms	1000	10:00
タスク2	2	20ms	500	20:00
タスク3	3	30ms	300	30:00
タスク4	4	40ms	200	40:00
タスク5	5	50ms	100	50:00

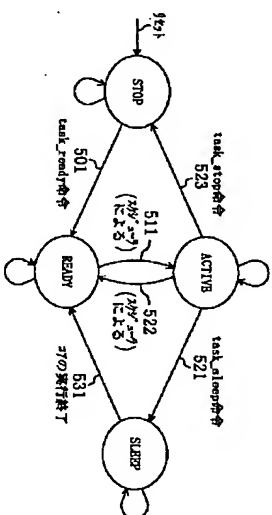
【図3】



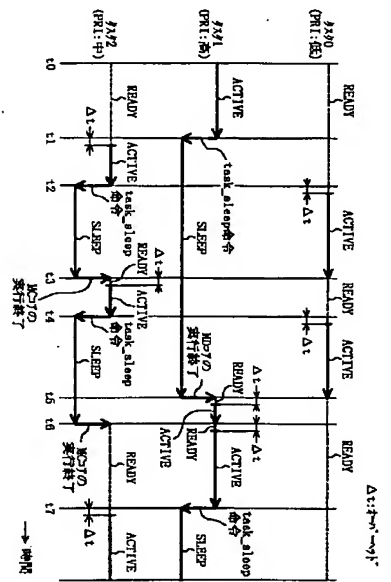
【図4】



【図5】



【図7】



フロントページの続き

(56) 参考文献 特開 平6-28323 (J.P. A)

インターフェース1995年1月号 (CQ

出版社)、P. 134~146

National Technica

l Report, Vol. 40, No.

6 (1994年12月)、P. 122~128 (J I

C S T 資料番号: G0474A)

(58) 調査した分野(Int. Cl. 7, DB 名)

G06F 9/46

G06F 15/16

G06F 9/38

G06T 1/00

H04N 7/24

JICSTスタイル (JOIS)

C SDB (日本国特許庁)